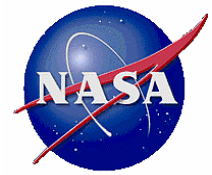
Software Integrity Level Assessment and Planning

Developing an Approach for Focusing Independent Verification and Validation Tasks

The Goals we had in Mind

- The approach to assigning a Software Integrity Level (SIL) to a software component is similar to the IV&V Analysis Levels (IALs) in the Criticality Analysis and Risk Assessment (CARA) process; basic, focused, etc
- The idea of SILs is based around the concept from IEEE 1012 and IEE 15504
- We have kept several goals in mind while building this process
  - Scalability – The process should be reasonably applicable from a mission level down to a function level
  - Components would be ranked based on risk – This would be a combination of criticality of the component and “Error Potential”
  - Only as complex as necessary – We want a relatively simple system that can be executed across a broad range of experience levels
  - Minimal impact to the development projects – The process should require a minimal level of participation from the project we are working with
  - Objective Criteria –This would help to keep the process simple as engineering judgment is reduced and measurements are used
  - The tasking approach would be different for each integrity level
  - Applicable throughout the life cycle
  - The outcome is understandable by the development project – The process and reasons for the results can be completely described in a planning and scoping report and will make sense to a general engineer/project manager



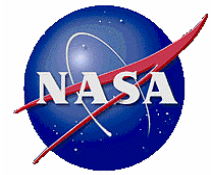
IV&V Facility

Software Integrity Levels (SILs)

- Development of a SIL scheme
  - The SIL scheme is generally a 4 level scheme from a low level of 1 to a high level of 4
    - Highest integrity is a 4, lowest integrity is a 1
    - Other options are being considered, but focus is on having 4 levels
- SIL Definitions (IEEE 1012-2004 draft)
  - 4 – Software element must execute correctly or grave consequences (loss of life, loss of system, economic or social loss) will occur. No mitigation is possible.
  - 3 – Software element must execute correctly or the intended use (mission) of the system/software will not be realized causing serious consequences (permanent injury, major system degradation, economic or social impact). Partial to complete mitigation.
  - 2 – Software element must execute correctly or an intended function will not be realized causing minor consequences. Complete mitigation possible.
  - 1 – Software element must execute correctly or intended function will not be realized causing negligible consequences. Mitigation not required.

Determining SILs from a Top-Down Approach

- Our focus in working from the top down created a convergence among several things that we do
- Since one of our goals was to rank the software by risk, we thought about using the 5 x 5 risk matrix as a tool in defining the risk
- In order to use that matrix we needed to define the axes of the matrix
- Nominally the axes are consequence and likelihood
  - Consequence is fairly straight forward and the definitions of the consequence should be covered by the severity level definitions (convergence between issue severity and criticality of components)
  - Likelihood is a little more complex and can be seen in several different ways
    - One is the probability of an error in the software being executed during any given time
    - Two is the likelihood that the developer will insert an error into the software
      - Does not imply that the error will manifest itself
    - Third, a combination of the two probabilities
  - The approach chosen was the second one as it focused more on prevention of errors being inserted rather than prevention of errors executing and fit better with our Agency objective of early detection of errors
  - We named this probability Error Potential

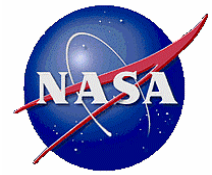


Defining the Axes

- Determining the axes to be used in the matrix is only part of the puzzle
- Each axis needs to have a good definition of each level within that access
- The Facility has generated a first draft of a table for Consequence and also for Error Potential
- The Consequence table is made up of three factors
  - Human Safety which is related to risks of death and injury
  - Asset Safety which concerns physical damage to hardware components of the mission to include critical assets that may be used to develop the components
  - Performance which concerns the ability of the component to affect mission success
- The Error Potential table is also made of three factors
  - Development Approach which is concerned with the organization and experience of the development team
  - Software Characteristics which is concerned with complexity, degree of innovation and size of the system
  - Development Process which is concerned with the state of the development process in terms of artifact maturity, reuse approach and formality of process

Consequence

	Severity				
	5	**4**	**3**	**2**	**1**
Human Safety	Loss of life	Injury or potential for injury		Discomfort or nuisance	
Asset Safety	Complete loss of vehicle/spacecraft/ system	Permanent loss of primary component	Permanent loss of non-primary component Complete loss of other critical asset	Partial loss of other critical asset	Short-term partial loss of other critical asset
Performance	Unable to achieve minimum mission/science objectives Unable to achieve minimum success criteria	Unable to achieve multiple mission/science objectives Loss of primary capability	Unable to achieve a particular primary mission/science objective Unrecoverable loss of primary mission/ science data	Unable to achieve non-primary mission/science objective Loss of non-primary capability	Unrecoverable loss of non-primary mission/ science data



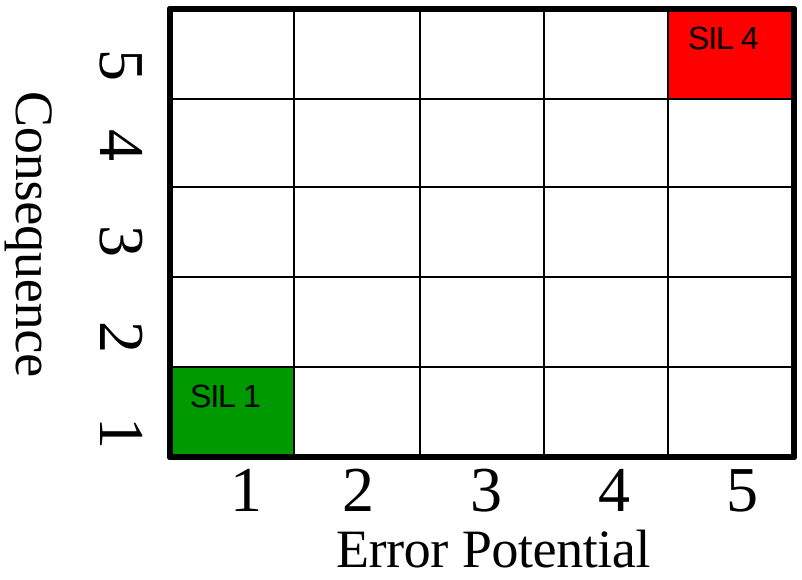
IV&V Facility

Error Potential

	Error Potential				
	5	**4**	**3**	**2**	**1**
Development Approach	Minimal domain or related experience (less than 5 years) Developer uses multiple subcontractors and management/staff that are not co-located (i.e., geographically dispersed)	Some domain or related experience (5-10 years) Developer uses one subcontractor and management/staff that are not co-located (i.e., geographically dispersed)	Nominal domain or related experience (10+ years) Developer does not use subcontractor and developer staff/management are not co-located	Developed one like system Developer does use subcontractor(s) and developer staff/management are co-located	Developed more than one like system or current incumbent Developer does not use subcontractors and developer staff/management are co-located
Software Characteristics	Multiple resource scheduling with dynamically changing priorities or distributed real-time control. Performance critical embedded system. Highly coupled dynamic relational and object structures. Object uses different end items (sensors) in different modes (stages) of system operation. Difficult and unstructured numerical analysis: highly accurate analysis of noisy, stochastic data and/or complex parallelization. Basic principles observed and reported -or- Technology concept and/or application formulated -or- Analytical & Experimental critical functions and/or characteristics proof-of-concept. More than 850K SLOC	Component and/or breadboard validation in a laboratory environment -or- Component and/or breadboard validation in relevant environment. 400K to 850K SLOC	Simple nesting with some inter-module control including decision tables, message passing and middleware supported distributed processing. Simple I/O processing including status checking and error processing. Multi-file input and single file input with minimal structural changes to the files. Function behaves differently in different modes of system operation. Standard math and statistical routines to include basic vector operations. System/subsystem model or prototype demonstration in a relevant environment. 50K to 400K SLOC	System prototype demonstration in a space environment -or- actual system complete and "Flight Qualified" through test and demonstration (ground or space). 10K to 50K SLOC	Straight line code with few to no nested structured programming operators: Dos, CASEs, IF THEN ELSEs. Simple module composition via procedure calls or simple scripts. Simple read-write statements with simple formats. Simple COTS-DB queries and updates. Function operates in only one more of system operation. Evaluation of simple expressions. Actual system "Flight Proven" through successful mission operations. Less than 10K SLOC
Development Process	Formality of process Re-use approach Artifcat maturity				

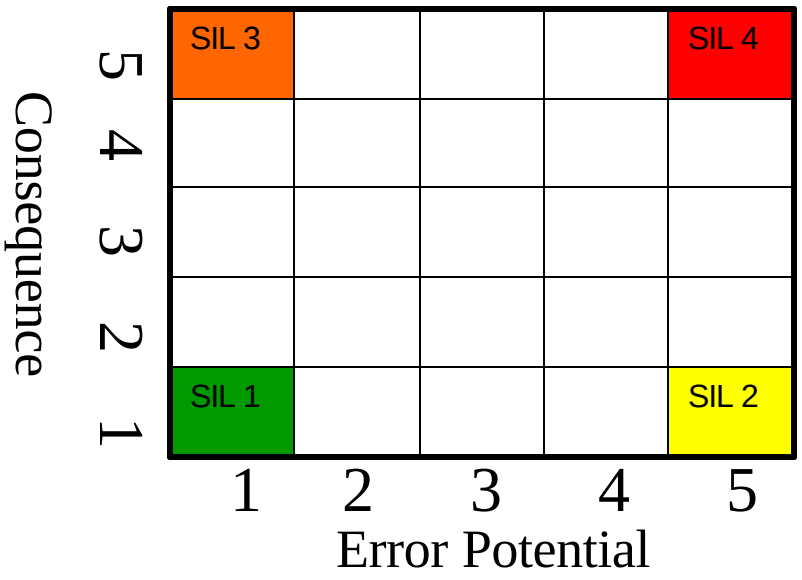
Fitting the SILs on a 5 x 5

- Once we determined the axes, we start to look at the 5 x 5 to determine how to categorize each box in to a SIL
  - A basic SIL 4 and 1 are the easiest to plot
  - With SIL 4 at the upper right and SIL 1 at the lower left
  - The exercise is the determining how to fill in the remaining boxes



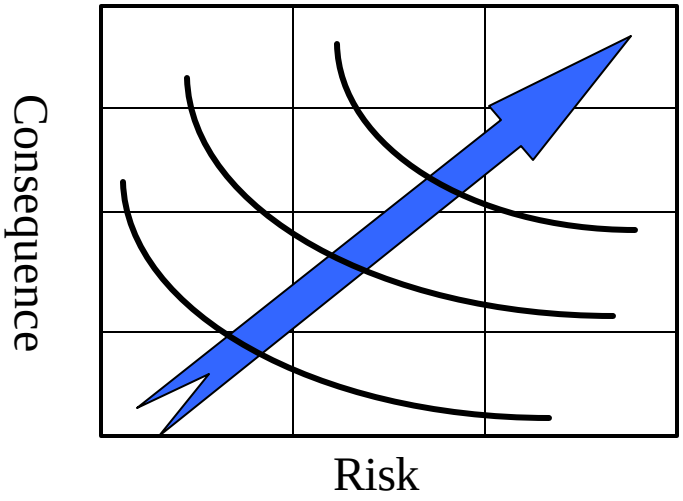
Where to put SIL 3 and SIL 2…

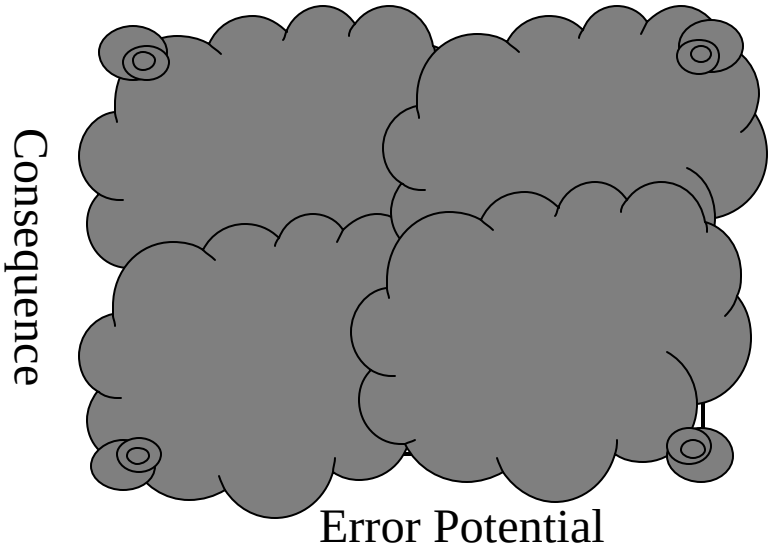
- The next decision was to determine which corner is SIL 2 and which is SIL 3
- The decision here was to use consequence as the deciding factor
- The consequence of an error seems to be more important even when it is low risk versus the likelihood of an error when the consequence is minimal
- This can be seen also in the SIL definitions from IEEE 1012 where mitigation is seen as a reduction in error potential (our view)



IV&V Facility

Some Thoughts on the Matrix Layout





- A CARA score can be broken into its components and plotted on a 4 x 3 matrix
- What you will see though is that components that score the same overall (i.e., have the same CARA score) can get that score in different ways
  - So does a 1 criticality, 3 risk demand the same set of tasks that a 3 criticality and a 1 risk does?
  - We thought about this and felt that the answer should be no, but CARA does not discriminate between the two
- We felt that something that had low consequence but high error potential should most likely a different set of analysis tasks compared to something that had high consequence but low error potential
- Our current approach is that low consequence but high error potential should have tasking focused on verification since we want to limit the number of errors
- Conversely, high consequence but low error potential would focus on validation since we want to make sure the software is the right software for the job
- Note that these will be relative to the final matrix distribution chosen and essentially define SILs 2 and 3 respectively

We still need to make a decision

- This is just the start of trying to determine how best to approach the job of determining how to categorize software components based on integrity levels
- We are confident in our approach of using the 5x5 matrix as a means of conveying the integrity information back to a project
- However, the implementation in how to layout the 5x5 matrix is still in work
- It is into this environment that we are interested in feedback from other government agencies and industry

So what about tasking?

- Indirectly tied to the definition of the SILs on the matrix is the definition of tasks for a given SIL
- A starting point would be to say that a SIL 4 gets a complete set of tasking
  - This means that a SIL 4 component would get the complete set of tasks for the given type of mission (human, robotic, instrument, data system)
- Conversely, a SIL 1 would get some minimal set of tasking (or perhaps nothing)
- So a question exists about the minimal set of tasking but perhaps another question needs to be answered first – Should some software on a NASA project not be analyzed by IV&V?
  - Or in a slightly different way – Should some Class A software on a NASA project not be analyzed by IV&V?
- If the answer to the question is yes, then SIL 1 is a strong candidate for being No IV&V
- If the answer to the question is no (i.e., all project software gets some IV&V analysis) then SIL 1 is a minimal task set

Some Additional Thoughts on Tasking

- The current approach to determining the task set
  - Start with a defined set of tasks for each “area” of the matrix
  - Additional tasks may also be chosen based on a certain specific characteristic(s) of the software
- Other options have been considered
  - One option is to take the Human, Robotic, Instrument and Data System break down as equivalent to the SIL and thus the assignment has already been completed
    - However, this also leads one to simply say that there are no SILs as defined by IEEE
    - SIL 4 is a manned mission, SIL 3 is a robotic mission, etc.
    - The approach could work, but the Facility does not have confidence in the approach at this time
- More work needs to be done to finalize these ideas

IV&V Planning

- In the past, the Facility used an Memorandum of Agreement with a project to denote interfaces and funding agreements
- IV&V tasking was then defined in a separate project plan
  - The use of two documents to define the IV&V project created confusion (Code Q IV&V Assessment, 2003)
- Based on the results from the assessment the IV&V Transition Team looked into how to combine the two documents into one comprehensive document
- This document is called the Independent Verification and Validation Plan (IVVP)
- It covers all aspects of the IV&V project from resources, team organization and development project interfaces to anomaly resolution, tasking and documentation control
- The implementation of the IVVP is currently ongoing with a completion date of the end of the Fiscal Year

Using the IVVP

A generic process for using the IVVP would be as follows

- The IV&V Facility would initiate contact with a development project prior to System Requirements Review (or equivalent)
- At this time, some initial coordination would take place and a rough IVVP would be baselined to indicate the upcoming activities and tasks involving the project
  - This is mostly a definition of the formulation phase of the IV&V effort
- At an appropriate point prior to Preliminary Design Review, the IV&V planning and scoping activity would take place
  - The output from this activity would be incorporated into the IVVP
- The IVVP is then updated as needed based on changes to the IV&V effort or the development project

Future Directions

- The matrix distribution needs to be finalized
- The tasking questions need to be answered
- Goal is to get input from the industry and possibly other Government institutions to help define and refine the approach
- Several options are being considered
  - Developing a survey to collect input on how to define the Error Potential
  - Holding a workshop with a goal to finish the workshop with a Peer Reviewed and useable process
- The IVVP work needs to be completed and the process for its use finalized